

踏波寻珠·潜水器定位探索

摘 要

针对问题一，需要综合考虑海底地形、海水密度、洋流、潜水艇物理特性。本文通过对海底地形的分析预测洋流并建立笛卡尔三维坐标系，利用牛顿运动方程和动力学方程，建立模拟潜水艇在失去信号情况下的运动轨迹模型，并根据实际运行中的表现调整模型参数，提高模型准确性。预测轨迹图见图4，图5。

针对问题二，需要探讨搜索设备选择，并分析搜索设备选择的可用性。为了直观地判断附加搜索设备的选择性，通过比较分析来量化收益和成本。效益-成本比(BCR)指标评估项目或投资计划的有效性，计算为总效益与总成本的比率。

针对问题三，需要预测初始部署点，搜索方式的选择。为了搜索设备更快搜索到潜水器，使用动态规划模拟商旅问题以作为初始部署点的选择，使用粒子群算法模拟搜索设备的搜索方式以使搜索更为高效。

针对问题四，需要使用相关性分析评价此模型更换区域后的可靠性，使用蚁群算法模拟多潜水器进而对商旅问题拓展。

最后对整个搜索模型进行总结，评价。

关键词： 牛顿运动方程；效益成本比；动态规划；粒子群算法；蚁群模型；

一、引言

1.1 问题背景

一家位于希腊的公司，致力于制造能够携带人类深入海底最深处的潜水艇。然而，由于下潜深度越大，信号丢失的概率也越大，压强逐渐升高，动力系统损坏的概率也越大。考虑到船上游客生命安全需要制定一定的安全措施，为此我们通过学习动力学相关知识，开发出一个模型用于预测潜水艇随时间的位置。

1.2 问题重述

考虑到问题的背景信息和约束条件，我们需要完成以下任务：

- **问题一：**1. 建立一个潜水艇运动轨迹模型用于预测潜水艇随时间的位置。
2. 分析预测模型的局限性
3. 分析潜水艇在事故前定期发送的信息以减少不确定性；潜水艇需要的设备。
- **问题二：**当考虑在母船上携带额外的搜索设备时，需要综合考虑设备的类型、可用性、维护成本、准备情况和使用成本等因素。使用成本效益模型计算出成本效益比可以判断携带那些设备。
- **问题三：**开发一个模型来推荐潜水器的初始部署点和搜索模式，以尽量减少潜水器丢失的定位时间。
- **问题四：**在其他一场或多场比赛中测试当前建立的模型，并评价该模型对于比赛中波动的预测结果；若预测结果欠佳，能否确定包含在未来模型中的因素？该模型在其他比赛、锦标赛、场地类型以及其他运动中普适性如何？

1.3 文献综述

伴随着人类探索、开发和利用海洋活动的大量开展和国际贸易及海上运输的发展，各种海难事故不断发生. 为了救助海难事故、打捞失沉设备、检查水下工程设施、考察水下物质, 从而挽救经济上的损失, 需要对水下各种目标物进行搜寻、探测和打捞, 而完成这些任务需要各种相关设备和工具. 人们常用大海捞针来形容做事的困难，可想而知在环境复杂的水下进行搜寻、探测的难度。水下搜寻、探测一般需先确定疑似区域再进行识别和确认，因此需各种设备进行相互配合，声学设备和地球物理设备适合大范围搜寻、探测，光学设备适合近距离观测和识别。无论采用哪种仪器设备进行水下搜寻、探测，都必须根据实际工作情况，按照仪器设备的原理和优缺点进行合理的选择，而且多种设备可以形成互补，同时应用能达到更好的搜寻、探测效果和经济效益。[\[1\]](#)

PsO算法（粒子群算法）是计算智能领域，除蚁群算法外的另外一种群智能算法。最早由Kennedy和Eberhart于1995年提出。它同遗传算法类似，通过个体间的协作和竞争实现全局搜索。系统初始化为一组随机解，称之为粒子。通过粒子在搜索空间的飞行完成寻优，在数学公式中即为迭代，它没有遗传算法的交叉以及变异算子，而是粒子在解空间追随最优的粒子进行搜索。PSO原理十分简单，所需参数也较少，并且易于实现。已经应用到很多的领域。[\[2\]](#)

旅行商问题作为组合优化研究中最具挑战的问题之一，自被提出以来就引起了学术界的广泛关注并提出了大量的方法来解决它。蚁群算法是求解复杂组合优化问题的一种启发式仿生进化算法，是求解旅行商问题的有效手段。本文分别介绍蚁群算法中几个有代表性的算法，综述了蚁群算法的改进、融合和应用的文献研究进展，以评价近年来不同版本的蚁群算法为解决旅行商问题的发展和研究成果，并针对改进蚁群算法结构框架、算法参数的设置及优化、信息素优化和混合算法等方面，对现被提出的改进算法进行了分类综述。对蚁群算法在未来对旅行商问题及其他不同领域的研究内容和研究热点的进一步发展提供了展望和依据。[3]

1.4 工作流程

为了便于描述和可视化，我们绘制了一个流程图（图1）来表示我们的工作。

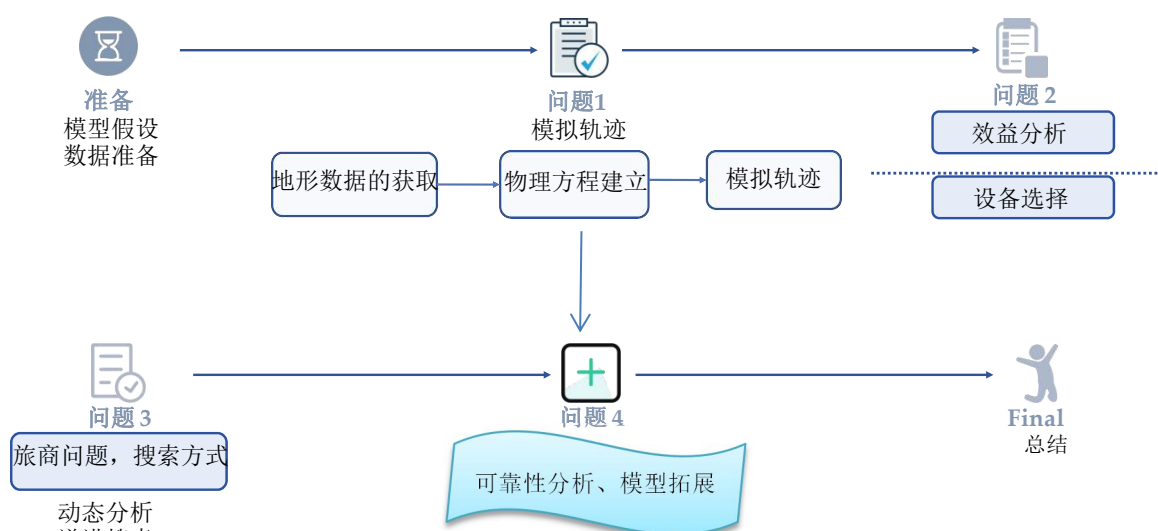


图 1 工作流程图

二、模型的准备

2.1 假设与说明

为了简化这个问题，我们做出了以下假设，每个假设都是适当合理的。

- 假设 1:** 潜水艇的运动不受海洋生物的影响，潜水艇下潜期间不会发生地震海啸等极端情况

<→ **说明:** 潜水艇行动缓慢，海底生物种类多样且活动存在较大的不可预测性难以纳入模型。极端天气不可预测且对潜水艇运行轨迹影响较大。
- 假设 2:** 假设潜水艇是一个规则的物体

<→ **说明:** 规则的物体方便进行受力分析，方便模型的建立。
- 假设 3:** 潜水艇不会撞击海底山体，潜水艇不会丧失上浮能力。

<→ **说明:** 潜水艇若撞击后可能会被卡住不能移动，那么模型预测将会失效。若失去上浮能力，潜水艇将一直在海水中运动预测难度太大。

2.2符号定义

表1列出了本文中使用的符号。有些变量在不同的地方有不同的含义，这里不一一列举，我们将在每一节中详细讨论。

表 1 符号定义

符号	定义
C_d	阻力系数
F_d	阻力
A	洋流对潜艇横截面的影响系数
N	总效益
P	总成本
η	效益成本比
r	相关系数

2.3数据收集

为了加强我们分析的可信度，除赛题给予的数据外，我们还从以下网站收集了信息，见表2。在此基础上，通过数理统计整理出海底地形数据，用于模型的建立。

表 2 数据来源网站及网址

来源网站	网址
GEONCO cle anet	https://download.gebco.net/ https://earth.nullschool.net/

三、建立牛顿运动方程的预测模型

3.1地形数据的获取

为了能够真实地进行潜艇轨迹预测，我们在爱奥尼亚海域圈出一块足够大的矩形区域作为模型预测的范围，我们在GEONCO下载了该矩形区域的海底地形图片。如图1为我们选定的矩形区域经纬度。

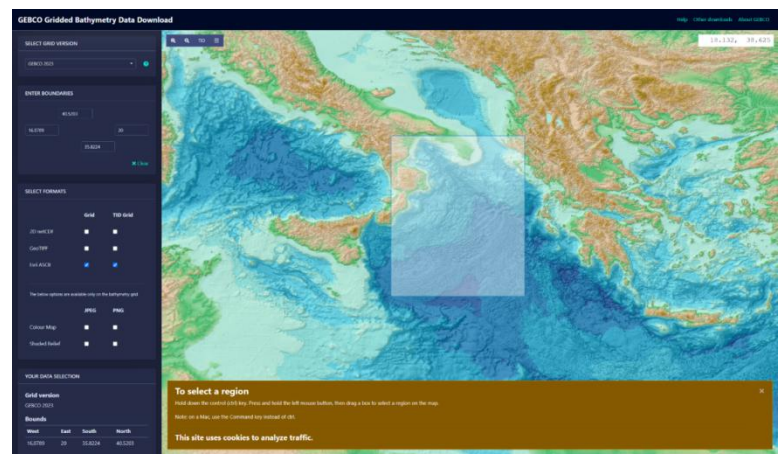


图1预测范围图

根据在CEONCO收集的海底地形图片，我们利用matlab将海底地形图转换为灰度图,假设陆地用较高值表示，海洋用较低值表示，设置海平面阈值，这里归一化为 0.5,并利用surf函数创建3D图，并使用颜色映射来表示海洋深度。如图2为海底地形图；如图3为海底地形3D图。

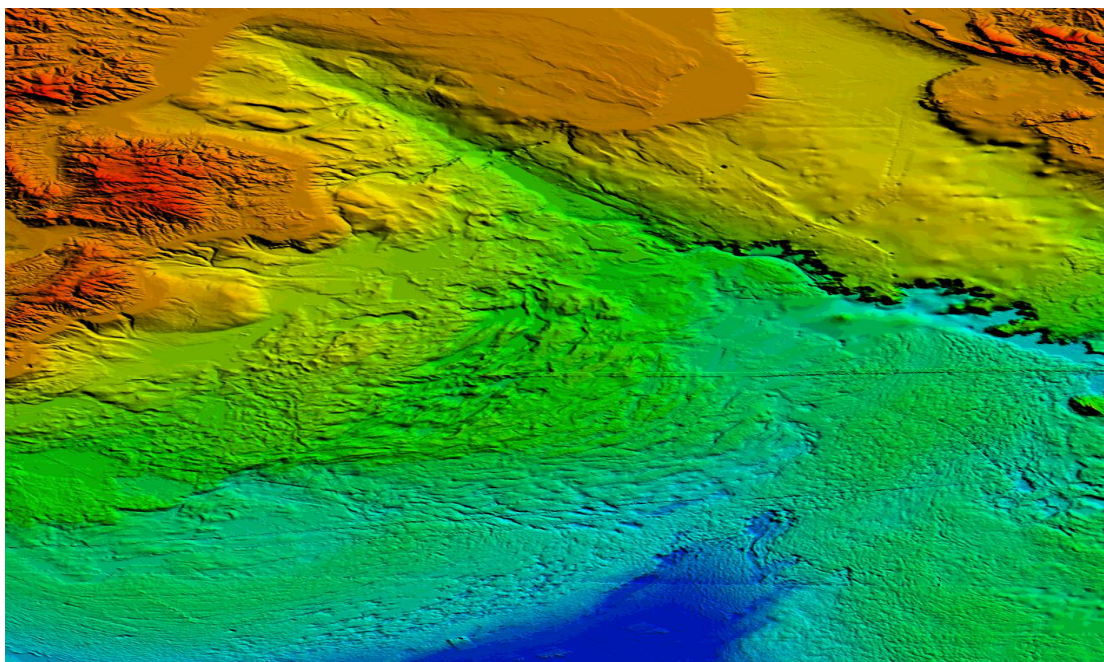


图2海底地形彩图

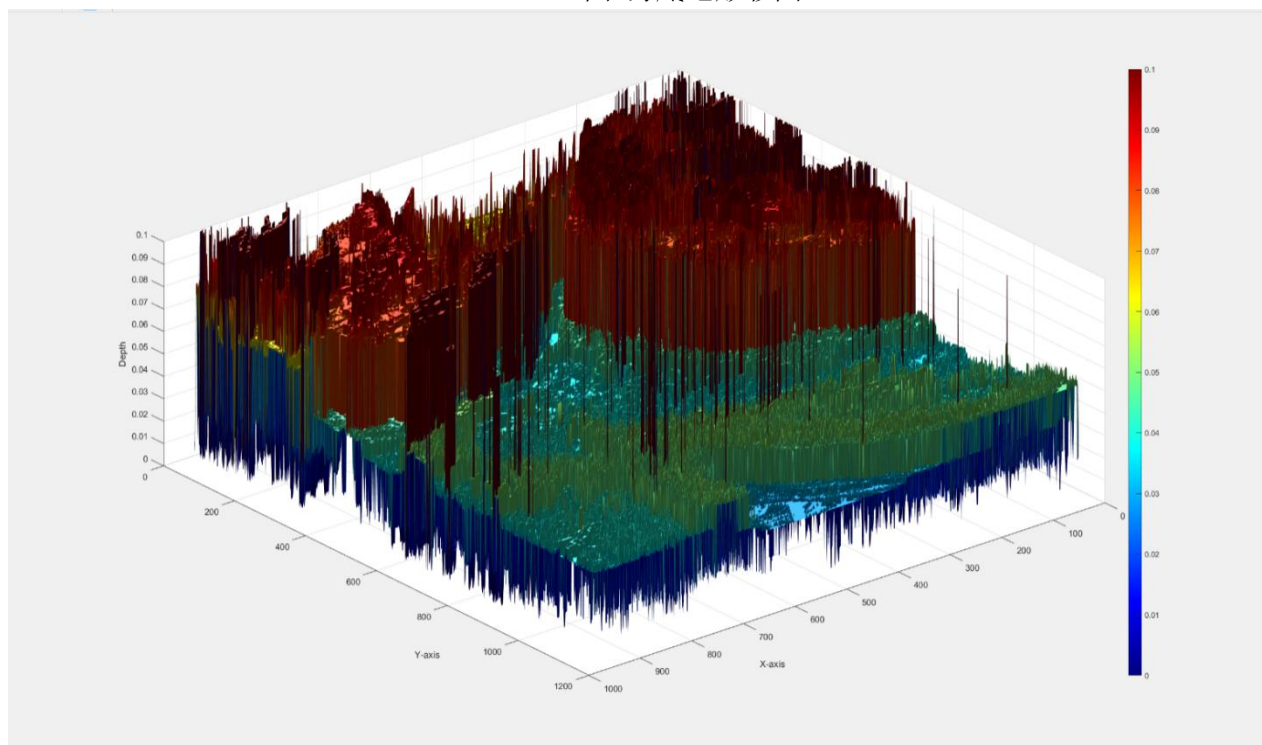


图3海底地形3D图

根据海底地形3D图，我们可以得到海底地形高度数据，我们主要利用该数据预测洋流，并建立预测模型。

3.2 模型涉及物理学方程及知识及模型的建立

1. 牛顿第二定律: $F=ma$ (1)

- F 是力, m 是物体的质量, a 是物体的加速度。这个方程描述了力与加速度之间的关系, 是牛顿运动方程的核心之一。

2. 阻力方程: $F_d = -C_d * A * \frac{1}{2} \rho v^2$ (2)

- F_d 是阻力, C_d 是阻力系数, A 是物体横截面积, ρ 是水的密度, v 是速度。这个方程描述了物体在流体中受到的阻力, 它与速度的平方成正比。

3. 浮力方程: $F_b = V_{submarine} \rho_{water} g$ (3)

- F_b 是浮力, 是潜艇的体积, $V_{submarine}$ 是水的密度, g 是重力加速度。这个方程描述了物体在液体中受到的向上浮力, 它等于排开液体的体积乘以液体的密度和重力加速度。

4. 重力方程: $F_g = mg$ (4)

- F_g 是重力, m 是物体的质量, g 是重力加速度。这个方程描述了物体受到的重力, 它与物体的质量成正比。

5. 水流对物体的横向力: $F_c = \rho_{water} A v_{current}^2 A_{submarine}$ (5)

- F_c 是水流对物体的横向力, ρ_{water} 是水的密度, $V_{current}$ 是水流速度, $A_{submarine}$ 是物体横截面积。这个方程描述了水流对物体在横向方向上施加的力, 它与水流速度的平方成正比。

计算上升力

$$F_{buoyancy} = \rho_{water} * V_{submarine} * g; \quad (6)$$

计算阻力

$$F_{drag} = 0.5 * C_d * \rho_{water} * V_{submarine} * v(i-1)^2; \quad (7)$$

计算洋流对潜艇的影响

$$F_{current} = 0.5 * \rho_{water} * A * V_{current} * v(i-1); \quad (8)$$

计算合力, 其中, 因为我们假设潜艇不会接触海底地形, 所以海底地形对潜艇的力可以忽略不记, 即 $F_{terrain} = 0$

$$F_{total} = F_{buoyancy} - F_{drag} - F_{current} - F_{terrain}; \quad (9)$$

根据合力在不同时间的变化函数, 我们得到了潜艇随时间变化的加速度函数, 进而可以建立潜艇随时间变化的运动轨迹模型。如图4为潜艇深度随时间变化图; 如图5为我们预测得到的潜艇预测轨迹图。

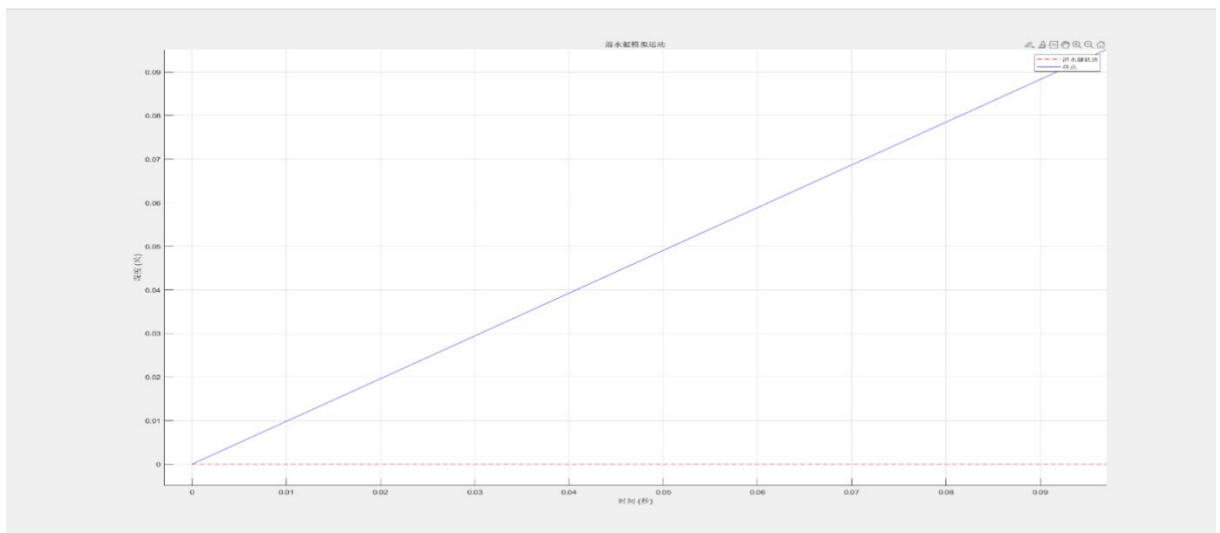


图4潜水艇深度变化图

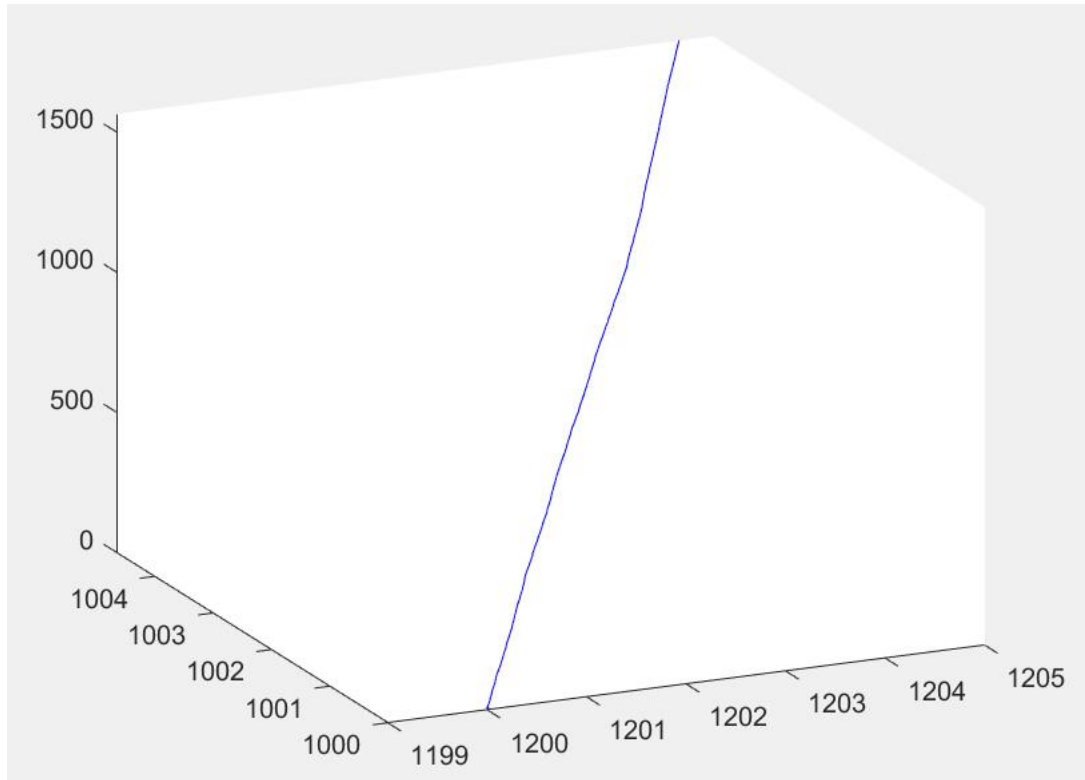


图5潜水艇预测轨迹图

3.3 预测模型的局限性

简化的洋流模型：模型假设洋流是恒定的，但实际情况下洋流可能会随着时间和位置而变化，可能受到季节性等因素的影响。因此，模型中使用的简化洋流模型可能无法准确地捕捉到真实环境中的复杂情况。

海底地形的精确性：模型中使用了简化的海底地形数据，但实际情况下海底地形可能会非常复杂，并且可能会随着位置的不同而变化。因此，模型中使用的简化地形数据可能无法准确地反映真实的海底地形，从而影响了模型的准确性。

忽略其他因素：模型中可能忽略了一些其他因素，如风力、潮汐等对洋流的影响，以及潜水艇本身的特性（如船体形状、推进系统等）。这些因素可能会对潜水艇在水中的运动产生重要影响，但在这个简化的模型中被忽略了。

3.4 分析潜水艇在事故前定期发送的信息以减少不确定性；潜水艇需要的设备。

为减少不确定性，我们需要考虑以下几个因素：

1. 潜水艇内在因素内旅客的身体情况及含氧量会影响潜水艇的运行状态，若有人出现不适或者含氧量过低潜水艇可能上浮至海面，影响潜水艇运行轨迹
2. 潜水艇周围环境因素，包括水温、压强、能见度。超过阈值范围的情况可能会造成潜水艇损坏、失联、动力缺失等情况

综合上述因素考虑，我们认为：

1. 潜水艇定期发送自身位置信息这要求潜水艇装备可靠的通信设备（如无线电）和配备定位系统（如GPS）；
2. 传感器：潜水艇需要各种传感器来监测船体状态、环境参数和船员生命迹象。这些传感器可能包括氧气传感器、水温传感器、压力传感器等。

四. 额外搜索设备选择

4.1 效益和成本的量化

在管理和经济学中，效益和成本的量化是非常重要的，它们通常用于决策分析、投资评估和资源分配。为了更直观判断额外搜索设备的选择性，可以对设备的效益和成本量化处理。量化可以使用比较分析的方法，将效益和成本与替代方案或基准进行比较，以便做出更合理的决策。比如，定义一个理想搜索设备，此搜索设备搜索半径为500米，现有一个搜索半径为800米的搜索设备，则此搜索设备的搜索范围效益为1.6。将效益与成本的量化如下：

$$a = \frac{\text{设备效益}}{\text{基准效益}} \quad (10)$$

$$A = \frac{\text{设备成本}}{\text{基准成本}} \quad (11)$$

4.2 效益成本比的计算

效益成本比（Benefit-Cost Ratio）是一种用于评估项目或投资方案的效益的指标，通常用于决策分析中。搜索设备有搜索范围、搜索精度、使用时间等效益，将这些效益设置不同比重和就是此搜索设备的总效益。总效益表达式如下：

$$N = \sum_{i=1}^n (\beta_1 a + \beta_2 b + \beta_3 c + \dots) \quad (12)$$

其中 β 表示各效益占总效益的权重且

$$\sum_{i=1}^n (\beta_1 + \beta_2 + \beta_3 + \dots + \beta_n) = 1 \quad (13)$$

总成本同理，将维护资金、购买资金[1]、老化等根据不同比重和即总成本。总成本表达式如下：

$$P = \sum_{i=1}^p (\omega_1 A + \omega_2 B + \omega_3 C + \dots) \quad (14)$$

其中 ω 表示各成本占总成本的权重且

$$\sum_{i=1}^p (\omega_1 + \omega_2 + \omega_3 + \dots + \omega_n) = 1 \quad (15)$$

效益成本比就是总效益与总成本的比值，成本效益比 = 总成本 / 总效益，表达式如下：

$$\eta = \frac{\sum_{i=1}^n (\beta_1 a + \beta_2 b + \beta_3 c + \dots)}{\sum_{i=1}^p (\omega_1 A + \omega_2 B + \omega_3 C + \dots)} \quad (16)$$

在计算效益成本比时，希望这个比值大于1，这表示项目或投资所带来的效益大于成本，是一个有利的投资选择。也可以设置一个基准效益成本比，如果设备的效益成本比大于基准效益成本比，则认为此设备可以携带。

4.3 预测模型的局限性

无法衡量的成本和收益：某些项目可能对环境造成负面影响，如自然资源的损耗、生态系统破坏等，一些项目可能对人们的健康和安全产生风险，如污染物排放、事故风险

等，难以衡量。

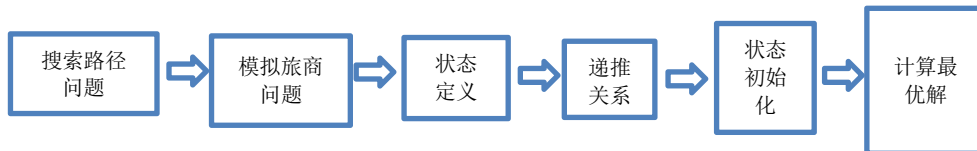
基准效益和基准成本设定的难度：设定基准效益和基准成本涉及对未来事件和结果的预测和估计。这些估计往往是主观的，并且可能受到不确定性的影响。不同的人可能有不同的意见和看法，从而导致基准的设定具有主观性和不确定性。设定基准效益和基准成本需要依赖可靠的数据支持。然而，在某些情况下，可能缺乏充分的数据或者数据的质量不够高，这会影响对基准的设定。设定基准效益和基准成本需要耗费时间和资源进行调研、分析和评估。在实际项目评估中，时间和资源往往是有限的，因此可能难以进行充分的数据收集和分析，从而影响基准的设定。

权重设定的难度：设定权重涉及对不同因素的重要性进行主观判断。不同的人可能会有不同的看法和偏好，导致权重设定具有主观性。在实际情况下，决策所涉及的因素可能非常复杂，而且彼此之间相互关联。在这种情况下，确定权重可能需要进行深入的分析和综合考虑，增加了权重设定的复杂性。

五、初始部署点的搜索模式的确认

5.1 初始部署点——动态规划模拟旅商问题 (TSP)

搜救时间的重要性在于对被搜救对象生命安全和搜救效果的直接影响。缩短时间的方法有多种，提前部署搜救力量和设备、最短路径的判断，路线对搜救速度的影响等。其中可以模拟旅商问题确定搜救最短路径以便搜索设备初始部署点的选择。旅行商问题是要找到一条路径，使得旅行商可以经过所有城市并且最后回到起始城市，同时要求路径的总长度最短。将搜索设备模拟为旅商，将问题一得出来的潜水器位置模拟为所有经过城市，初始部署点模拟为起始城市，搜索设备的搜救过程就转化成为了旅商问题。



其中递推关系使用**动态规划**方法[1]，假设从顶点s出发，令 $d(i, V)$ 表示从顶点i出发经过V(是一个点的集合)中各个顶点一次且仅一次，最后回到出发点s的最短路径长度。推导:(分情况来讨论)

①当V为空集，那么 $d(i, V)$ ，表示直接从i回到s了，此时 $d(i, V) = G_s$ 且 $(i \neq s)$

②如果V不为空，那么就是对子问题的最优求解。你必须在V这个城市集合中，尝试每一个，并求出最优解。

$$d(i, V) = \min(C_i + d(k, V - \{k\})) \quad (17)$$

注:ck表示选择的和城市i的距离， $d(k, V - \{k\})$ 是一个子问题。

综上所述，TSP问题的动态规划方程如下：

$$d(i, V) = \begin{cases} c_{is}, & V = \emptyset, i \neq s \\ \min\{c_{ik} + d(k, V - \{K\})\}, & k \in V, V \neq \emptyset \end{cases}$$

其中 s 为起点。

5.2 递进搜索——粒子群算法

另一种节约搜救时间的方式就是正确选择和改进搜索设备的搜索方式，搜索方式多种多样，例如扇形搜索、网格搜索、螺旋搜索、雅克布式搜索、数据驱动搜索等，本次使用递进搜索方式模拟潜水器的搜索，递进搜索过程如下：

①定义搜索区域和初始位置：搜索区域使用问第一地形灰度图得出的3D地形，初始位置使用模拟旅商问题确定的初始部署点。

②定义目标位置：目标位置使用问题一预测的潜水艇轨迹。

③递进式搜索：从初始位置开始，逐步扩大搜索范围，直到搜索到目标位置或达到最大搜索范围为止。在每次搜索中，通过循环遍历当前搜索范围内的所有位置，并记录搜索到的位置。如果搜索到了目标位置，则标记找到目标并终止搜索。

④绘制搜索位置图：将搜索到的位置以及目标位置绘制在三维坐标系中，以便直观地查看搜索过程。

绘制的搜索位置图和搜索轨迹图如下：

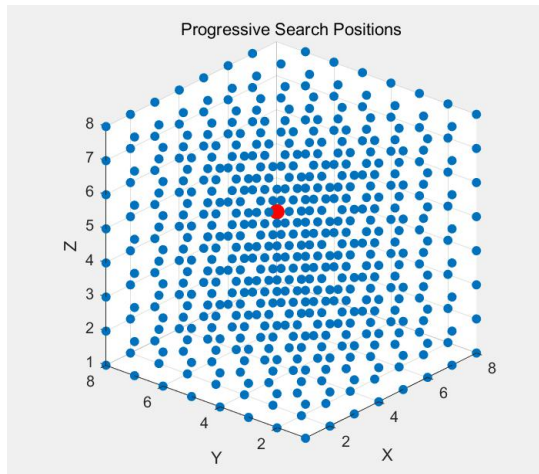


图6搜索位置图

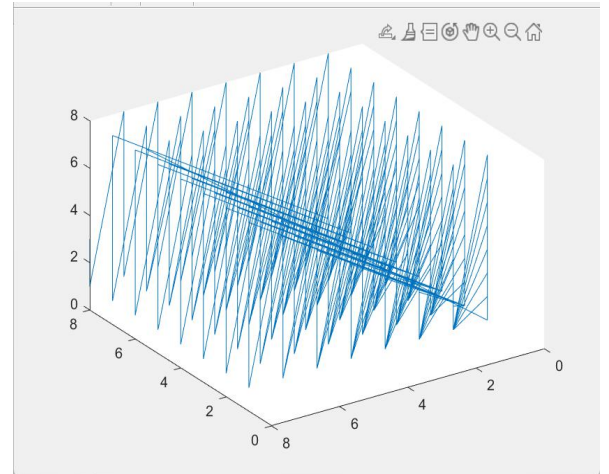


图7搜索轨迹图

5.3 预测模型的局限性

计算复杂性：TSP是一个NP难题，意味着随着城市数量的增加，问题的计算复杂性呈指数级增长。动态规划虽然可以在某些情况下对小规模问题进行有效求解，但对于大规模问题，计算复杂性会变得非常高，耗费大量时间和计算资源。

局部最优解：动态规划方法在解决TSP时可能会陷入局部最优解，尤其是在存在大量城市并且问题结构复杂的情况下。这可能导致无法找到全局最优解，而只能获得次优解。

空间需求：递进搜索需要存储和管理大量的中间状态和搜索信息，特别是在搜索空间扩展时。这将对内存和存储资源提出挑战，尤其是在处理大规模问题时。

六、模型可靠性，旅商问题拓展

6.1 模型可靠性——相关性分析

为了检验潜水器在不同海域位置模拟的可靠性，使用相关性分析进行评价。其中模拟潜水器Z轴上浮时，采用恒定浮力和重力产生改变速度的效果，潜水器Z轴上浮位移是时间的二次函数，所以不再对Z轴轨迹进行相关性分析。

相关系数定义为：

$$\rho_{X,Y} = \frac{\text{cov}(X,Y)}{\sigma_X \sigma_Y} = \frac{E[(X - EX)(Y - EY)]}{\sigma_X \sigma_Y} = \frac{E(XY) - E(X)E(Y)}{\sqrt{E(X^2) - E^2(X)} \sqrt{E(Y^2) - E^2(Y)}}$$

同样，首先找到加勒比海的地形图，进行灰度转化为地形数据，带入问题一所需代码求出潜水器运动轨迹，然后对x和y的运动轨迹进行相关性分析。其中两个海域的x,y相关性分析及结果如下

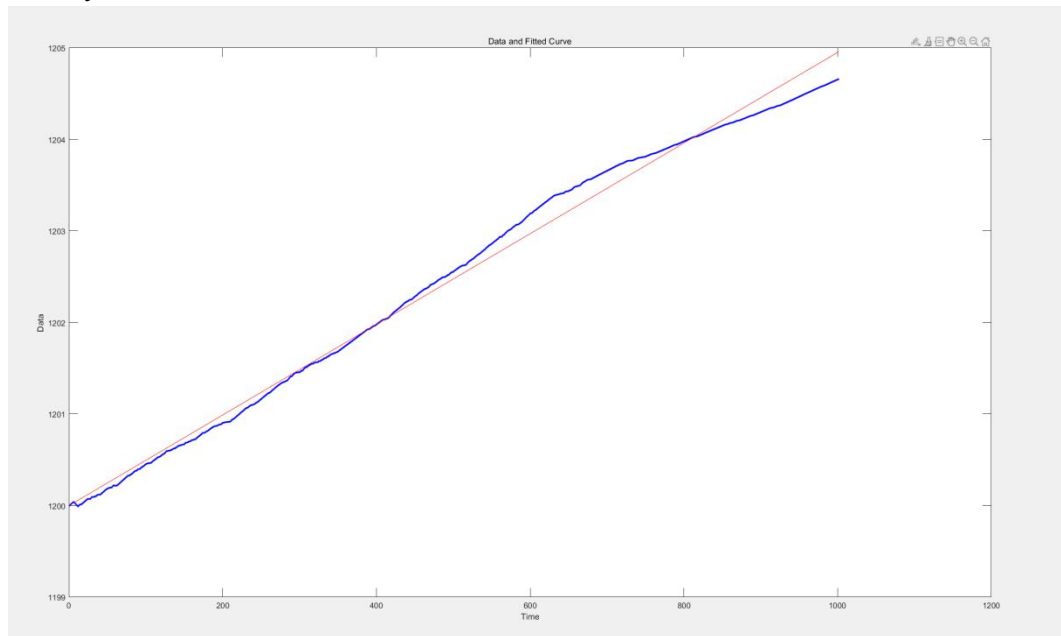


图8爱奥尼亚海的x轴轨迹相关性分析

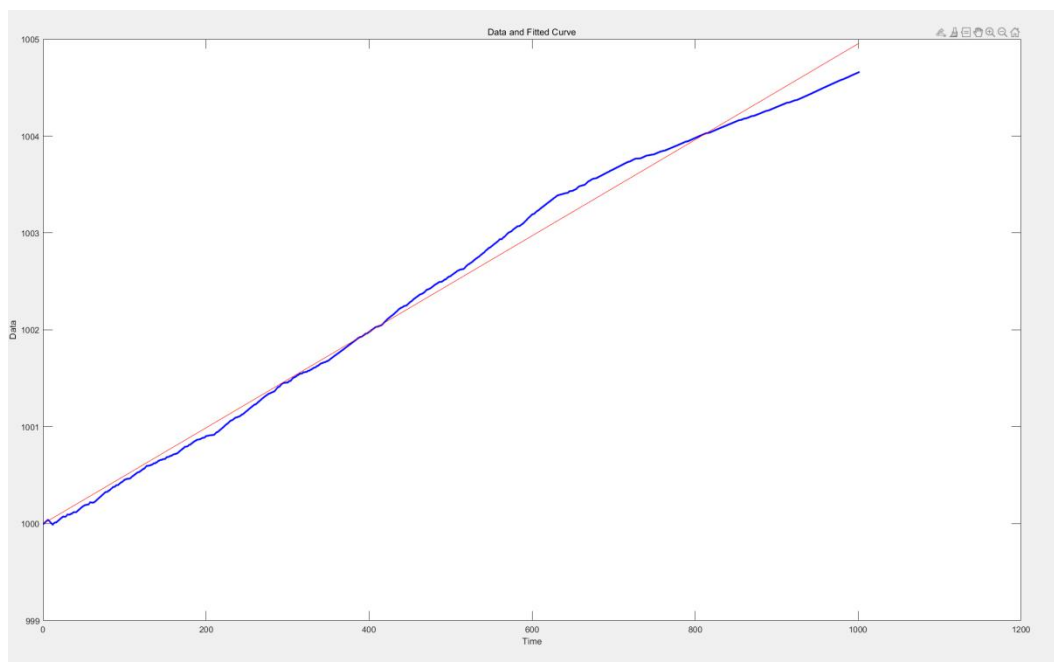


图9爱奥尼亚海的y轴轨迹相关性分析

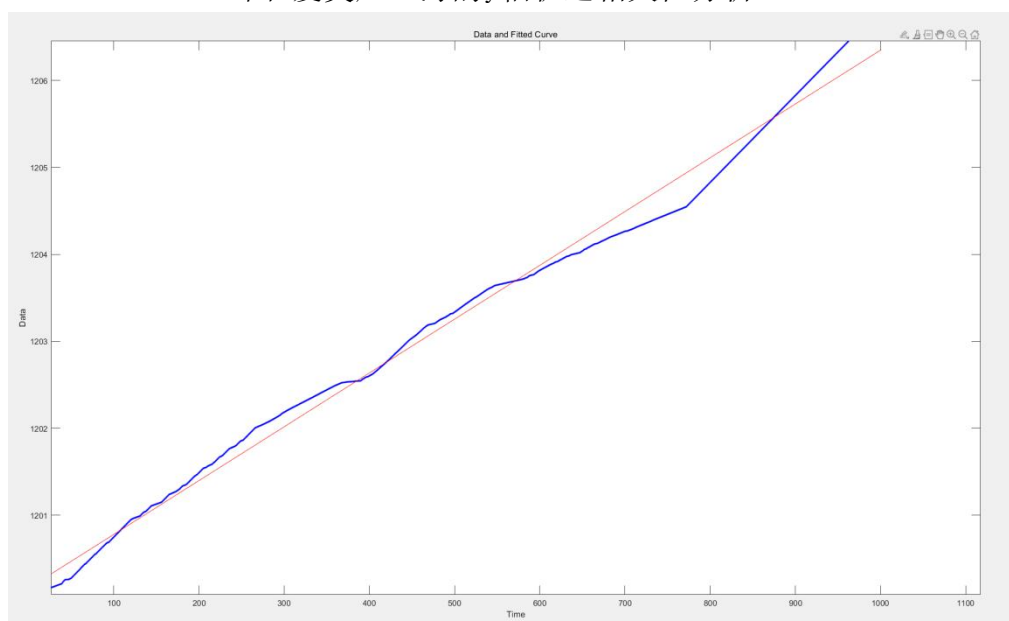


图10加勒比海的x轴相关性分析

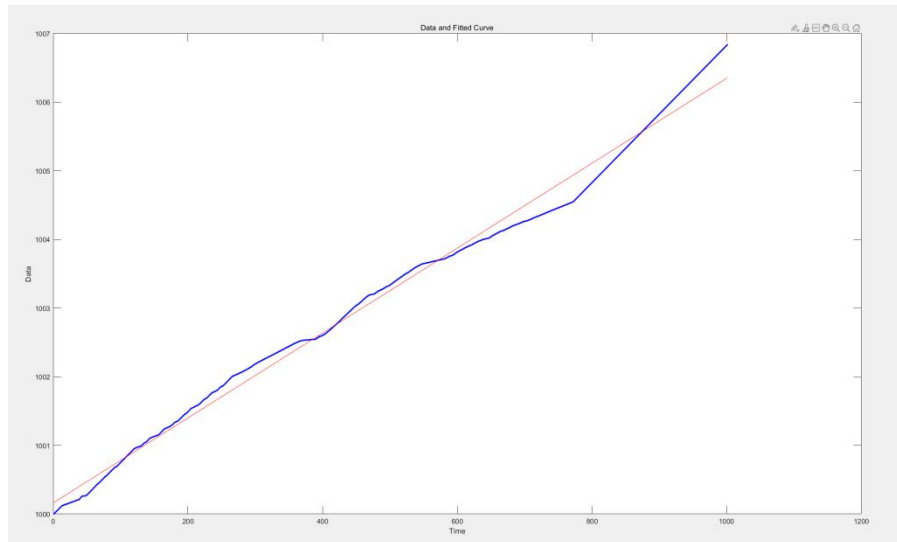


图11加勒比的y轴相关性分析

6.2多潜水器问题——模拟蚁群算法

与问题三旅商问题不同，本题考虑到多潜水器的最短路径问题，蚁群算法可以将多潜水器模拟为蚁群，蚁群通过随机探路，最终找到城市，而城市就是指坏掉的潜水器。

将多潜水器搜索过程通过使用蚁群算法的过程如下图12：

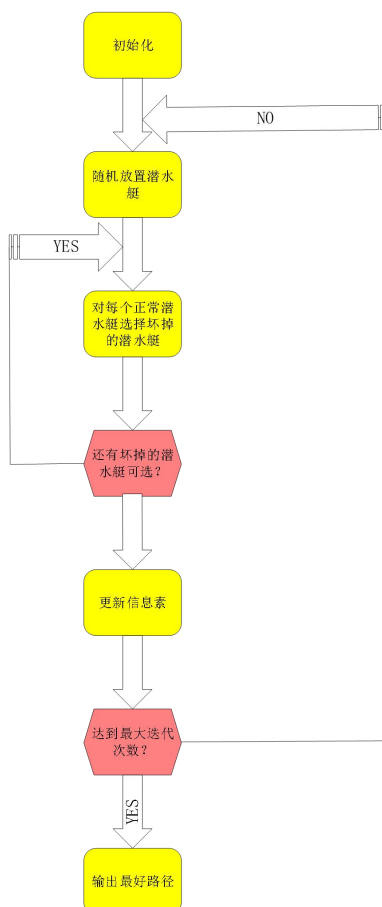


图12 模拟蚁群算法流程图

通过以上过程模拟一次蚁群算法得出轨迹图13如下

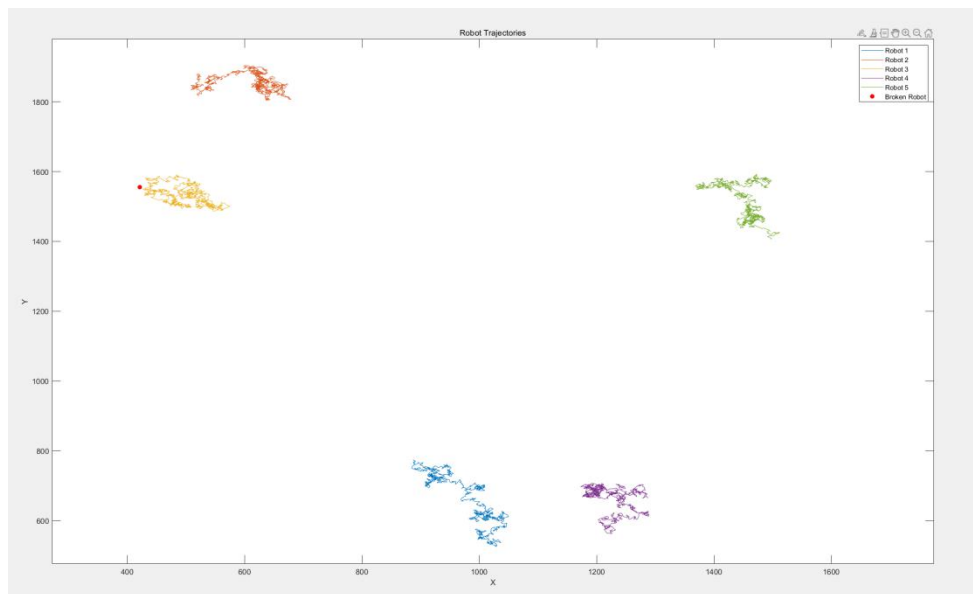


图13模拟蚁群轨迹

七、模型的评价

7.1 模型优点

- 本模型通过两个不同区域对比分析，突出区域兼容性高。
- 本模型采用最基本的牛顿力学，易于使用，易于改进。
- 递进搜索方式范围广泛，不易留下搜寻死角。
- 通过相关性分析结果，使模型更加可靠。

7.2 模型缺点与改进

- 此模型假设的是均匀洋流力场，实际洋流是三维不均匀立场，可将模型各个点的洋流进行一次更新。
- 因为潜水器损坏的不确定性，多潜水器问题中可设置蚂蚁和城市的单向转化，用来潜水器在潜水过程损坏，进而被其它正常潜水器发现。

八、总结

首先使用找到的爱奥尼亚海洋地形图进行灰度转换得出地形数据图，建立海洋下的牛顿力学方程，将地形数据带入方程，并模拟一个洋流可预测潜水器轨迹。针对设备选择问题，使用效益成本比，通过对比效益成本比的大小可直观的判断设备种类的选择。初始部署点和搜索方式简化为最短路径问题，分别使用动态规划和粒子群算法求解。最后再预测一次潜水器在加勒比海运动轨迹，通过对两个海域轨迹进行相关性分析检验模型的可靠性，最后使用蚁群模型拓展最短路径问题。总体过程为轨迹预测、效益成本分析、初始部署点和搜索方式的确立、可靠性检验和最短路径问题拓展。

参考文献

- [1] 浅谈水下目标搜寻与探测设备[C]. //第九届中国国际救捞论坛 论文集. 2016:81-82..
- [2] 权文,王晓丹,王坚,等. 粒子群算法的新改进思想-递进式搜索[J]. 计算机工程与应用,2009,45(2):44-47. DOI:10.3778/j.issn.1002-8331.2009.02.012.
- [3] 郭城成,田立勤,武文星. 蚁群算法在求解旅行商问题中的应用综述[J]. 计算机系统应用,2023,32(3):1-14. DOI:10.15888/j.cnki.csa.008976

附录

[A] 支撑文件列表

支撑文件列表及结构如下（按文件夹进行分类）：

文件夹名	描述
Code	解决问题所有源程序及环境要求
Figures	论文中所有矢量图示
Data	解决问题所用数据及生成数据
Results	程序输出结果

<ul style="list-style-type: none"> Code <ul style="list-style-type: none"> diyiwen.m dixingtu.m dierwen.m disanwendijinsousuo.m disiwenxiangguanxing.m disiwenyiqun.m、 tsp_3d_dynamic_programming.m tsptest.m Data <ul style="list-style-type: none"> 第一问爱奥尼亚海底地形数据.xlsx 第四问加勒比海地形数据.xlsx 第四问加勒比z数据.xlsx 第四问加勒比y数据.xlsx 第四问加勒比x数据.xlsx 第四问爱奥尼亚z数据.xlsx 第四问爱奥尼亚y数据.xlsx 第四问爱奥尼亚x数据.xlsx 	<ul style="list-style-type: none"> Figures <ul style="list-style-type: none"> 爱奥尼亚坐标.png 奥埃尼亚海地形彩图.png 加勒比海地形彩图.png 加勒比海坐标.png Results <ul style="list-style-type: none"> 爱奥尼亚x数据拟合.png 爱奥尼亚y数据拟合.png 加勒比x数据拟合.png 加勒比y数据拟合.png 爱奥尼亚地形图.png 加勒比地形图.png 搜索轨迹图.png 搜索位置图.png 蚁群模型结果图.png
--	---

[B]使用的软件、环境

B.1: 为解决该问题，我们所使用的主要软件有：

- WPS Office 教育考试专用版
- MATLABR2021a
- MATLABR2023a
- EXCEL2022
- Global Mapper

[C] 问题解决源程序

C.Task One1.1

```
% 模拟参数
V_submarine = 30; % 潜水艇体积 (立方米)
rho_water = 1025; % 海水密度 (千克/立方米)
g = 9.81; % 重力加速度 (米/秒^2)
A_submarine = pi * 1 * 1.5; % 潜水艇横截面积 (假设为椭圆形, 单位: 平方米)
Cd = 0.5; % 潜水艇的阻力系数
A = 0.8; % 洋流对潜水艇的横截面积影响系数
v_current = 0.1; % 洋流速度 (假设为一个常数, 单位: 米/秒)

% 导入海底地形图数据
terrain_data = ['F:\桌面\Date\aiaroniyahaididixingz.xlsx'];

% 深度比例尺
depth_scale = 100; % 深度比例尺 (假设为100, 单位: 米)

% 时间参数
t_total = 100; % 总时间 (秒)
dt = 0.1; % 时间步长 (秒)
t = 0:dt:t_total; % 时间数组

% 初始条件
z0 = 0; % 初始深度 (假设在海平面上)
v0 = 0; % 初始速度

% 初始化数组
z = zeros(size(t));
v = zeros(size(t));
x = 1200; % 保存潜水艇在x方向的位置
y = 1000; % 保存潜水艇在y方向的位置

% 模拟运动
z(1) = z0;
v(1) = v0;
for i = 2:length(t)
    % 计算上升力
    F_buoyancy = rho_water * V_submarine * g;

    % 计算阻力
```

```

F_drag = 0.5 * Cd * rho_water * A_submarine * v(i-1)^2;

% 计算洋流对潜水艇的影响（模拟洋流沿海底的水平速度）
if i <= size(terrain_data, 1) && z(i-1) > 0
    v_x_current = v_current * interp1(1:size(terrain_data, 2), terrain_data(i, :), z(i-1) /
depth_scale, 'linear', 'extrap');
else
    v_x_current = v_current; % 当潜水艇到达海平面时，只受洋流作用
end
if i <= size(terrain_data, 1) && z(i-1) > 0
    v_y_current = v_current * interp1(1:size(terrain_data, 2), terrain_data(i, :), z(i-1) /
depth_scale, 'linear', 'extrap');
else
    v_y_current = v_current; % 当潜水艇到达海平面时，只受洋流作用
end
% 获取当前深度处的海底高度
if i <= size(terrain_data, 1)
    terrain_height = interp1(1:size(terrain_data, 2), terrain_data(i, :), z(i-1) / depth_scale,
'linear', 'extrap') * depth_scale;
else
    terrain_height = 0; % 超出海底地形数据的范围时，假设海底高度为0
end

% 计算海底地形对阻力的影响
if z(i-1) < terrain_height
    F_terrain = Cd * rho_water * A_submarine * g * (terrain_height - z(i-1));
else
    F_terrain = 0;
end

% 计算合力
F_total = F_buoyancy - F_drag;

% 计算加速度
a = F_total / (rho_water * V_submarine);

% 计算速度和深度
v(i) = v(i-1) + a * dt;
z(i) = max(0, z(i-1) + v(i) * dt); % 深度不允许小于0

% 更新潜水艇在x方向的位置

```

```

x(i) = x(i-1) + v_x_current * dt;

% 更新潜水艇在y方向的位置（假设潜水艇只做直线运动，不考虑侧向位移）
y(i) = y(i-1) + v_y_current * dt;

% 当潜水艇到达海平面时，停止模拟
if z(i) == 0 && z(i-1) > 0
    break;
end
end

% 绘制结果
plot3(x, y, z, 'b-');
xlabel

```

Task One1.2

```

% Load the image
img = imread('F:\桌面\5.11\DISIWEN88.png');
% Convert the image to grayscale if it is not already
if size(img, 3) == 3
    img = rgb2gray(img);
end
% Convert to double for processing
img = double(img);
% Normalize the image to get the intensity values between 0 and 1
img = (img - min(img(:))) / (max(img(:)) - min(img(:)));
% Assuming that land is represented by the higher values and sea by lower values,% set
the threshold for sea level, which is normalized to 0.5 in this case sea_level_threshold =
0.5; % This needs to be adjusted based on your image data % Set land (values above the
sea level threshold) to NaN so they are not plotted img(img > sea_level_threshold) = NaN;
% Generate x and y coordinates based on image size
[x, y] = meshgrid(1:size(img,2), 1:size(img,1));
% Create the 3D plot using the surf function
figure;
surf(x, y, img, 'EdgeColor', 'none'); % The land will not be plotted
% Adjust the view angle for better visualization
view(45, 30);
% Add labels and title if needed
xlabel('X-axis');
ylabel('Y-axis');
zlabel('Depth');

```

```

% Use the 'jet' colormap for the ocean depth to replicate the rainbow color distribution
colormap(jet);
% Add a color bar to indicate the scale
colorbar;
% Optional: Adjust lighting to improve the appearance
camlight left;
lighting phong;
% Save the plot to a file
saveas(gcf, '3D_ocean_floor_plot_with_jet_colormap.png')

```

Task Two 2.1

```

w1=0.2,w2=0.3,w3=0.5;
A=100;
B=300;
C=400;
r1=0.15;
r2=0.4;
r3=0.45;
a=100;
b=200;
c=400;
% 计算成本和收益
cb = w1*A+w2*B+w3*C;
sy = r1*a+r2*b+r3*c;
% 计算成本收益比
psy = sy/cb;
% 输出成本收益比
fprintf(1,'成本效益比=',psy),disp(psy);

```

Task Three 3.1.1

```

function [min_dist, min_path] = tsp_3d_dynamic_programming(points)
    n = size(points, 2);
    all_sets = 1:2^n;
    dp = inf(n, 2^n);
    dp(1, 1) = 0;

    for mask = 1:2^n
        for current_node = 1:n
            if bitget(mask, current_node) == 0
                continue;
            end

```

```

    prev_mask = mask - 2^(current_node-1);
    if prev_mask == 0
        continue;
    end
    for prev_node = 1:n
        if bitget(prev_mask, prev_node) == 0
            continue;
        end
        new_dist = dp(prev_node, prev_mask) + norm(points(prev_node, :) -
points(current_node, :));
        dp(current_node, mask) = min(dp(current_node, mask), new_dist);
    end
end
end

min_dist = inf;
for i = 2:n
    min_dist = min(min_dist, dp(i, 2^n-1) + norm(points(i, :) - points(1, :)));
end

% Reconstruct the path
current_node = 1;
current_mask = 2^n - 1;
min_path = [1];
for i = 1:n-1
    next_node = -1;
    for j = 1:n
        if bitget(current_mask, j) == 1 && (next_node == -1 || dp(j, current_mask) +
norm(points(j, :) - points(current_node, :)) < dp(next_node, current_mask) +
norm(points(next_node, :) - points(current_node, :)))
            next_node = j;
        end
    end
    min_path = [min_path, next_node];
    current_mask = current_mask - 2^(next_node-1);
    current_node = next_node;
end
min_path = [min_path, 1]; % Return to the start node
end

```

Task Three 3.1.2


```

% 测试动态规划解决三维旅行商问题的函数，并绘制图形

% 读取测试数据
points = rand(10, 3); % 随机生成10个三维点

% 调用函数计算最短路径距离
min_distance = dynamic_programming_3d_tsp(points);

% 显示结果
disp(['最短路径距离: ', num2str(min_distance)]);

% 绘制图形
figure;
plot3(points(:,1), points(:,2), points(:,3), 'bo', 'MarkerSize', 10); % 绘制点
hold on;

% 绘制最短路径
path_order = greedy_tsp(points); % 假设这里使用了贪婪算法找到了近似最短路径
path_points = points(path_order, :);
plot3(path_points(:,1), path_points(:,2), path_points(:,3), 'r-', 'LineWidth', 2); % 绘制路径

xlabel('X');
ylabel('Y');
zlabel('Z');
title('3D TSP Solution');
legend('Points', 'Shortest Path');
hold off;

```

Task Three 3.2

```

% 定义搜索区域的大小和初始位置
grid_size = [1001, 1001, 1001]; % 三维网格大小 [x, y, z]
initial_position = [1, 1, 1]; % 初始位置

% 定义目标位置
target_position = [8, 8, 3]; % 目标位置

% 初始化存储搜索位置的矩阵
search_positions = zeros(0, 3); % 初始为空

% 递进式搜索

```

```

search_range = 1; % 初始化搜索范围
found_target = false; % 标记是否找到目标
while search_range <= max(grid_size)
    % 在搜索范围内搜索
    for x = max(1, initial_position(1)-search_range):min(grid_size(1),
initial_position(1)+search_range)
        for y = max(1, initial_position(2)-search_range):min(grid_size(2),
initial_position(2)+search_range)
            for z = max(1, initial_position(3)-search_range):min(grid_size(3),
initial_position(3)+search_range)
                % 在当前位置执行搜索操作，这里可以根据具体情况进行操作

                disp(['Searching at position (', num2str(x), ', ', num2str(y), ', ', num2str(z), ')']);

                % 记录搜索到的位置
                search_positions = [search_positions; x, y, z];

                % 检查是否找到目标
                if isequal([x, y, z], target_position)
                    found_target = true;
                    break;
                end
            end
        end
    end
    if found_target
        break;
    end
end

% 如果找到目标，则终止搜索
if found_target
    break;
end

% 扩大搜索范围
search_range = search_range + 1;
end

```

```

% 绘制位置图
scatter3(search_positions(:, 1), search_positions(:, 2), search_positions(:, 3), 'filled');
hold on;
scatter3(target_position(1), target_position(2), target_position(3), 100, 'r', 'filled'); % 标记
目标位置
hold off;
xlabel('X');
ylabel('Y');
zlabel('Z');
title('Progressive Search Positions');
axis equal;
plot3(search_positions(:, 1), search_positions(:, 2), search_positions(:, 3))
% 显示是否找到目标
if found_target
    disp('目标已找到! ');
else
    disp('未找到目标。 ');
End

```

Task Four 4.1

```

% 读取Excel文件
data = xlsread('F:\桌面\5.11\diseiwen2y.xlsx', 'Sheet1', 'A1:ALM1');

% 计算相关系数
correlation_coefficient = corr(data);

% 拟合一次函数
x = 1:size(data, 2); % x轴数据
y = data; % y轴数据
coefficients = polyfit(x, y, 1); % 一次多项式拟合

% 计算拟合曲线
fit_line = polyval(coefficients, x);

% 绘制原始数据和拟合曲线
figure;
plot(x, y, 'b.', x, fit_line, 'r-');
xlabel('Time');
ylabel('Data');
title('Data and Fitted Curve');

```

```
% 打印相关系数和拟合参数
disp(['相关系数r= ', num2str(correlation_coefficient)]);
disp(['一次函数系数: ', num2str(coefficients)]);
```

Task Four 4.2

```
% 设置参数
num_robots = 5; % 水下机器人数量
search_radius = 50; % 搜索半径
num_iterations = 1000; % 迭代次数
evaporation_rate = 0.1; % 信息素蒸发率
pheromone_init = 1; % 初始信息素浓度
pheromone_max = 10; % 信息素最大值

% 初始化机器人位置和轨迹
robots_x = rand(1, num_robots) * 2000; % 海域宽度为100米
robots_y = rand(1, num_robots) * 2000; % 海域长度为100米
robot_trajectories = zeros(num_robots, 2, num_iterations);

% 初始化信息素矩阵
pheromones = ones(2000, 2000) * pheromone_init;

% 坏掉的水下机器人位置
broken_robot_x = rand() * 2000;
broken_robot_y = rand() * 2000;

% 迭代搜索
for iter = 1:num_iterations
    % 水下机器人搜索
    for i = 1:num_robots
        % 计算当前机器人到其他机器人的距离
        distances = sqrt((robots_x - robots_x(i)).^2 + (robots_y - robots_y(i)).^2);

        % 找出搜索范围内的其他机器人
        neighbors = find(distances <= search_radius & (1:num_robots) ~= i);

        % 如果当前机器人在搜索半径内找到了坏掉的机器人，则朝着坏掉机器人移动
        if any(neighbors)
            next_x = broken_robot_x;
            next_y = broken_robot_y;
        else
            % 计算信息素浓度
```

```

    pheromone_values = pheromones(round(robots_x(i)), round(robots_y(i)));

    % 选择下一个搜索位置
    [~, next_index] = max(pheromone_values);
    next_x = robots_x(i) + rand() * 10 - 5; % 在搜索范围内随机移动
    next_y = robots_y(i) + rand() * 10 - 5; % 在搜索范围内随机移动
end

% 更新机器人位置和轨迹
robots_x(i) = next_x;
robots_y(i) = next_y;
robot_trajectories(i, :, iter) = [robots_x(i), robots_y(i)];

% 更新信息素浓度
pheromones(round(robots_x(i)), round(robots_y(i))) = min(pheromone_max,
pheromones(round(robots_x(i)), round(robots_y(i))) + 1);
end

% 信息素蒸发
pheromones = pheromones * (1 - evaporation_rate);
end

% 绘制机器人的运动轨迹
figure;
for i = 1:num_robots
    plot(squeeze(robot_trajectories(i, 1, :)), squeeze(robot_trajectories(i, 2, :)));
    hold on;
end

% 绘制坏掉的机器人位置
scatter(broken_robot_x, broken_robot_y, 'r', 'filled');

xlabel('X');
ylabel('Y');
title('Robot Trajectories');
axis([0 2000 0 2000]);
legend('Robot 1', 'Robot 2', 'Robot 3', 'Robot 4', 'Robot 5', 'Broken Robot');
hold off;

```
